

Mat1062: Computational Methods for PDE

Mary Pugh

April 8, 2008

Burger's Equation

It's clear how to use spectral methods for any linear constant coefficient partial differential equation on a periodic domain. We now turn to nonlinear equations. As a sample equation, we will consider Burger's equation with dissipation:

$$u_t + uu_x = Du_{xx} \quad \text{on } (0, L)$$

with periodic boundary conditions. The first step is to find a spectral representation of this problem. We seek $u(x, t)$ so that

$$\langle u_t + uu_x - Du_{xx}, \phi_\ell \rangle = 0 \quad \forall \ell \in \mathbb{Z}, \forall t > 0$$

for each basis function $\phi_\ell = \exp(i\ell 2\pi x/L)$. That is,

$$\frac{d}{dt} \hat{u}_\ell + \widehat{(uu_x)}_\ell + D\ell^2 \frac{2\pi^2}{L} \hat{u}_\ell = 0$$

with initial data $\hat{u}_\ell(0) = \hat{u}_{0\ell}$. And so again, we have reduced the PDE to infinitely many ODE but they are no longer decoupled:

$$\begin{aligned} u(x, t) &= \sum_{\ell=-\infty}^{\infty} \hat{u}_\ell(t) e^{i\ell 2\pi x/L} \implies u_x(x, t) = \sum_{\ell=-\infty}^{\infty} i\ell \frac{2\pi}{L} \hat{u}_\ell(t) e^{i\ell 2\pi x/L} \\ u(x, t)u_x(x, t) &= \sum_{\tilde{m}=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} im \frac{2\pi}{L} \hat{u}_m(t) \hat{u}_{\tilde{m}}(t) e^{i(\tilde{m}+m)2\pi x/L} \\ \implies \widehat{(uu_x)}_\ell &= \sum_{m=-\infty}^{\infty} im \frac{2\pi}{L} \hat{u}_m(t) \hat{u}_{\ell-m}(t) \end{aligned}$$

And so the infinite system of ODE is

$$\begin{cases} \frac{d}{dt} \hat{u}_\ell = -D\ell^2 \left(\frac{2\pi}{L}\right)^2 \hat{u}_\ell(t) - \sum_{m=-\infty}^{\infty} im \frac{2\pi}{L} \hat{u}_m(t) \hat{u}_{\ell-m}(t) \\ \hat{u}_\ell(0) = \hat{u}_{0\ell} \end{cases} \quad (1)$$

for all $\ell \in \mathbb{Z}$.

Time-Stepping

Assuming that we somehow figure out how to deal with the convolution sum in (1), we see that in spectral space we will need to solve a nonlinear ODE at each wave number ℓ . And so we start by considering the ODE

$$\frac{d}{dt}u(t) = f(u(t)), \quad u(0) = u_0.$$

This is the first time we've needed to timestep a nonlinear ODE. Up until now, all of our ODE were linear because the PDE they came from were linear.

Clearly, we can still easily implement explicit Euler:

$$u^{n+1} = u^n + kf(u^n).$$

However, if we want to implement either the fully implicit or the Crank-Nicolson scheme

$$u^{n+1} - kf(u^{n+1}) = u^n \quad u^{n+1} - \frac{k}{2}f(u^{n+1}) = u^n + \frac{k}{2}f(u^n)$$

then solving for u^{n+1} is no longer an exercise in linear algebra — we would need to use some nonlinear solver like Newton-Raphson iteration. Suddenly, Crank-Nicolson is no longer cheap to execute. If you're going to go through the hard work of coding up that nonlinear solver, you may wish to use it on a time-stepping scheme with higher accuracy. There is the third-order Adams-Moulton method:

$$u^{n+1} = u^n + \frac{1}{12}k(5f(u^{n+1}) + 8f(u^n) - f(u^{n-1}))$$

and the fourth-order Adams Moulton method:

$$u^{n+1} = u^n + \frac{1}{24}k(9f(u^{n+1}) + 19f(u^n) - 5f(u^{n-1}) + f(u^{n-2})).$$

To implement these schemes we need to invert one of $I - kf$ or $I - k/2 f$ or $I - 5k/12 f$ or $I - 9k/24 f$. Inverting any one of these will involve the same amount of labour coding and testing the code.

Because of the bother involved in programming up the nonlinear solver, and the slow-down caused by having to do a nonlinear solve with each time step, we may also wish to consider higher-order explicit schemes. The good news is that we can find such schemes and so there's no hard work in solving for u^{n+1} . The bad news is that these explicit schemes all have stability

constraints on the time-step k , just as the explicit Euler scheme does. Here is the second-order Adams-Bashforth method:

$$u^{n+1} = u^n + \frac{1}{2}k(3f(u^n) - f(u^{n-1})),$$

the third-order Adams-Bashforth method:

$$u^{n+1} = u^n + \frac{1}{12}k(23f(u^n) - 16f(u^{n-1}) + 5f(u^{n-2})),$$

and the fourth-order Adams-Bashforth method:

$$u^{n+1} = u^n + \frac{1}{24}k(55f(u^n) - 59f(u^{n-1}) + 37f(u^{n-2}) - 9f(u^{n-3})).$$

Note that all the higher-order Adams-Moulton and Adams-Bashforth schemes require memory of past values. This does not cause implementation problems. If you want to implement the third-order scheme then you would use explicit Euler to generate u^1 from u^0 . You'd then use the second-order scheme to generate u^2 from u^1 and u^0 . You'd then be in a position to start using the third-order scheme to generate u^n for all $n \geq 3$. Even though there were a few initial steps of lower-order accuracy, the total scheme would still be third-order accurate.

Alternate higher-order explicit schemes would include Runge-Kutta time-stepping methods. These are also explicit and, unfortunately, have stability constraints on the time-step. Runge-Kutta schemes do not use past values of u , they compute intermediate values; sort of like how the ADI Peaceman-Rachford scheme had to find an intermediate u in the process of finding u^{n+1} .

And so, given a particular problem that you want to compute you have to decide whether you are willing to spend the time programming up the nonlinear solver and have a code that runs a little more slowly because it has to execute a nonlinear solve at each time step or if you want to code up an explicit scheme which is easy to code and runs more quickly per time step except that because of the stability constraint you'll have to take more time steps to reach your desired end time.

For some problems, the stability constraint is prohibitive. For example, consider the Kuramoto-Sivashinsky equation: $u_t = -u_{xxxx} - u_{xx} - uu_x$ on $[0, L]$. This equation is a classic model problem for weak turbulence and the study of how PDE are related to dynamical systems. The solutions have very interesting behaviour, strongly influenced by the domain size. You need to be cunning about how you do the time-stepping for such a high-order equation: a naive explicit method will have a time step constraint of $k < ch^4$.

For dispersive problems, how one chooses the time-stepping is also influenced by questions of wanting to minimize the phase error introduced by the scheme.

Computing the Convolution Sum

In the system (1) for each $\ell \in \mathbb{Z}$ we need to compute

$$\sum_{m=-\infty}^{\infty} \hat{f}_m \hat{g}_{\ell-m} \quad (2)$$

where $\hat{f}_m = im(2\pi/L)\hat{u}_m$ and $\hat{g}_{\ell-m} = \hat{u}_{\ell-m}$. At first sight, this looks prohibitive: for a given ℓ we need to find

$$\cdots + \hat{f}_{-2} \hat{g}_{\ell+2} + \hat{f}_{-1} \hat{g}_{\ell+1} + \hat{f}_0 \hat{g}_{\ell} + \hat{f}_1 \hat{g}_{\ell-1} + \hat{f}_2 \hat{g}_{\ell-2} \cdots \quad (3)$$

Assume that ℓ is in the “active spectrum” at time t . Specifically, assume that $|\ell| \leq N_{as}$ where $\hat{f}_m \sim 10^{-16}$ for all $|m| > N_{as}/2$. Then the sum is over $N_{as} + 1$ terms:

$$\hat{f}_{-N_{as}/2} \hat{g}_{\ell+N_{as}/2} \cdots + \hat{f}_{-1} \hat{g}_{\ell+1} + \hat{f}_0 \hat{g}_{\ell} + \hat{f}_1 \hat{g}_{\ell-1} \cdots \hat{f}_{N_{as}/2} \hat{g}_{\ell-N_{as}/2}$$

The subscript is for “active spectrum” — N_{as} will be a function of time. Similarly, assume $\hat{g}_m \sim 10^{-16}$ for all $|m| > N_{as}/2$ (this is certainly true for the above choice of f and g). And so the sum is over $N_{as} - |\ell| + 1$ terms:

$$\widehat{(fg)}_{\ell} = \sum_{m=\max\{-N_{as}/2, \ell-N_{as}/2\}}^{\min\{N_{as}/2, \ell+N_{as}/2\}} \hat{f}_m \hat{g}_{\ell-m}.$$

Computing this sum is $\mathcal{O}(N_{as})$. And so we see that if we were to compute the convolution sum (2) for each ℓ in a direct and naive manner it would take $\mathcal{O}(N_{as}^2)$ operations. Which is prohibitively slow.

So how do we compute the convolution? We like to be in frequency space when it comes to derivatives — they’re diagonal operators there. And they’re harder to deal with in real space — we get into finite differences. On the other hand, we like to be in real-space when it comes to nonlinearities — if we want to compute u^2 at $x = x_j$ it’s just $(u(x_j))^2$. Nonlinearities are harder to deal with in frequency space because of the convolutions they entail. And so we want a scheme that does derivatives in frequency space and nonlinearities in real space. This is what is meant by a *pseudospectral* schem.

If we want to do explicit Euler time-stepping on Burger's equation:

$$\hat{u}_\ell^{n+1} = \hat{u}_\ell^n - k \left(-D\ell^2(2\pi/L)^2 \hat{u}_\ell^n - \widehat{(u^n u_x^n)}_\ell \right) \quad (4)$$

we would do it as follows.

1. At time t_n we are in spectral space and have the discrete Fourier coefficients $\{\hat{u}_0^n, \dots, \hat{u}_{N-1}^n\}$. We use these Fourier coefficients to create

$$\hat{v}_k = \begin{cases} ik \frac{2\pi}{L} \hat{u}_k^n & 0 \leq k \leq \lfloor N/2 \rfloor - 1 \\ 0 & k = \lfloor N/2 \rfloor \\ -i(N-k) \frac{2\pi}{L} \hat{u}_k^n & \lfloor N/2 \rfloor + 1 \leq k \leq N-1 \end{cases}$$

which are a good approximation of the discrete Fourier coefficients of u_x^n (if N is large enough for u to be spectrally resolved).

2. Apply the inverse discrete Fourier transform (ifft) to $\{\hat{u}_\ell^n\}$ and $\{\hat{v}_\ell\}$ to recover u^n and u_x^n in physical space. Now apply the nonlinearity, creating

$$w_j = (\text{ifft}(\hat{u}^n))_j (\text{ifft}(\hat{v}))_j, \quad j = 0, \dots, N-1$$

which is uu_x in physical space.

3. Apply the discrete Fourier transform (fft) to $\{w_j\}$ creating

$$\widehat{(u^n u_x^n)}_\ell = \text{fft}(w)_\ell \quad \ell = 0, \dots, N-1$$

Now that we have $\widehat{(u^n u_x^n)}_\ell$ we can take a time step via (4)

This method for computing the nonlinearity can introduce aliasing error. And so if you ever find yourself on the verge of using a spectral method in your research I strongly encourage you to look into how to deal with this aliasing error. You can learn more in §3.4 of Canuto et al.

Let's compute!

We want to test various time-stepping schemes. First, we would like to have an exact solution to compare our numerics to. Such exact solutions are

easily created, thanks to the Cole-Hopf transform¹. Specifically, if ϕ is a solution of the heat equation $\phi_t = D\phi_{xx}$ and

$$u = -D \frac{\phi_x}{\phi}$$

then u is a solution of $u_t + uu_x = Du_{xx}$.

Using this, for the periodic problem on $[0, L]$ we take $\phi(x, t)$ to be any linear combination of basic solutions such as

$$e^{-D\ell^2(\frac{2\pi}{L})^2 t} \sin(\ell 2\pi x/L)$$

and use this to generate a solution $u(x, t)$. If we were to take a single basic solution rather than a linear combination, the resulting $u(x, t)$ would be independent of time.

Specifically, we take

$$u_0(x) = -2D \frac{\cos(x)}{3 + \sin(x)} \implies u(x, t) = -2D \frac{e^{-Dt} \cos(x)}{3 + e^{-Dt} \sin(x)}$$

on $[0, 2\pi]$. We consider four time-stepping schemes:

1. An integrating factor scheme:

$$\hat{u}_\ell^{n+1} = e^{-D\ell^2 k} \left(\hat{u}_\ell^n - k \widehat{(u^n u_x^n)}_\ell \right)$$

2. Explicit Euler time-stepping:

$$\hat{u}_\ell^{n+1} = (1 - D\ell^2 k) \hat{u}_\ell^n - k \widehat{(u^n u_x^n)}_\ell$$

3. A scheme for which the linear term is treated implicitly and the non-linear term is treated explicitly:

$$\hat{u}_\ell^{n+1} = \frac{\hat{u}_\ell^n - k \widehat{(u^n u_x^n)}_\ell}{1 + D\ell^2 k}$$

4. Explicit second-order Adams-Bashforth:

$$\hat{u}_\ell^{n+1} = \hat{u}_\ell^n + \frac{k}{2} \left(-D\ell^2 (3\hat{u}_\ell^n - \hat{u}_\ell^{n-1}) - 3\widehat{(u^n u_x^n)}_\ell + \widehat{(u^{n-1} u_x^{n-1})}_\ell \right)$$

¹J.D. Cole "On a quasilinear parabolic equation occurring in aerodynamics" *Quarterly of Applied Mathematics* 9(1951)225-236, E. Hopf, "The partial differential equation $u_t + uu_x = \mu u_{xx}$ " *Communications in Pure and Applied Mathematics* 3(1950)201-230.

We test by taking $N = 128$ and the time step $k = 1/1000$. We compute up to time $T = 1/100$ and then compare the computed solution to the exact solution. We divide the time step by 2 and then repeat. Computing seven solutions in this way, we compare the ratios of the errors in the L^∞ norm and find that the first three schemes have ratios going to 2 and the fourth scheme has ratios going to 4, as expected.

We now take the time step $k = 1/64000$ and compare the schemes. We find

$$\begin{aligned} \text{integrating factor} &\implies \|\text{err}\|_{L^\infty} = 9.49e - 7 \\ \text{explicit} &\implies \|\text{err}\|_{L^\infty} = 7.79e - 7 \\ \text{mixed} &\implies \|\text{err}\|_{L^\infty} = 1.66e - 6 \\ \text{Adams-Bashforth} &\implies \|\text{err}\|_{L^\infty} = 1.33e - 9 \end{aligned}$$

The second-order Adams-Bashforth scheme has the smallest error, unsurprisingly.

The integrating factor and mixed schemes have no stability constraint on their time-step while the explicit and Adams-Bashforth scheme do. Indeed, when computing the explicit scheme, it was unstable for $k = 1/1000$ and $k = 1/2000$ but stable for $k = 1/4000$. The Adams-Bashforth scheme was unstable for $k = 1/4000$ too — it was stable for $k = 1/8000$. And so we see, in this example at least, that the Adams-Bashforth scheme had a tighter stability constraint than the explicit scheme.

You can find the “stability domains” for various time-stepping schemes in pretty much any graduate numerical analysis book, including the Canuto et al. book. Separately, stability analysis is usually done for *linear* systems which is one reason why it’s important to have some computational way of telling if your nonlinear computation is becoming unstable.

Nonlinear Schrödinger Equation

This is a paradigm example of a nonlinear wave equation. It arises physically in a variety of contexts, ranging from nonlinear optics (index of refraction depends on light intensity) to dynamics of superfluid helium. It illustrates a lot of interesting behavior,² and is a stellar example of spectral methods.

Loosely speaking, the equation comes in two varieties: *defocusing* or with a *repulsive* potential, and *focusing*, or with an *attractive* potential.

²See Catherine Sulem and Pierre-Louis Sulem, *The Nonlinear Schrödinger Equation: Self-Focusing and Wave Collapse*, Applied Mathematical Sciences Vol. 139, Springer 1999.

The difference in the equation is just a change of sign, but the behavior of solutions is very different. We start with the defocusing case.

Defocusing The PDE for the complex function $u(x, t)$ is

$$i u_t = u_{xx} - |u|^2 u.$$

In higher dimensions, the derivative just becomes the Laplacian. We shall suppose that the equation is to be solved on a periodic domain, $[0, L]$ in one dimension, or $[0, L]^d$ in d dimensions. This called the *cubic nonlinearity*; another popular choice is $|u|^{2\sigma} u$ with $\sigma = 1, 2, \dots$.

One way to understand this equation (as always) is to look at conserved quantities, of which two are of special importance. The *particle number* is

$$N(t) = \int |u(x, t)|^2 dx$$

(you may normalize it however you like). If the domain were unbounded (no periodicity), then we would impose decay conditions at ∞ so this quantity would be finite. To calculate the derivative, we recall that $|u|^2 = u \bar{u}$, so

$$\begin{aligned} \frac{dN}{dt} &= \int (u \bar{u}_t + u_t \bar{u}) dx \\ &= \int_0^L \left(u (i \bar{u}_{xx} - i |u|^2 \bar{u}) + \bar{u} (-i u_{xx} + i |u|^2 u) \right) dx \\ &= \int_0^L \left(i (-u_x \bar{u}_x + u_x \bar{u}_x) + i |u|^2 (-u \bar{u} + \bar{u} u) \right) dx \\ &= 0, \end{aligned}$$

where we have integrated by parts once on the derivative terms. Thus N is a constant of the motion.

The other conserved quantity is the energy, or *Hamiltonian*

$$H(t) = \int \left(|\nabla u|^2 + \frac{1}{2} |u|^4 \right) dx$$

(for general σ the second term is $|u|^{2\sigma+2}/(\sigma+1)$). Similar calculations as above show that $H(t)$ is also conserved.

Note that this energy is very similar to the one we used in the Allen-Cahn equation, except that there the time dynamics was *gradient descent*: it decreased the energy as rapidly as possible. Here the dynamics is that of a *Hamiltonian system*, that moves “across” the energy.

For the 1-D NLS with cubic nonlinearity, there is in fact an infinite family of conserved quantities, which leads to the study of “integrable systems.” In general, though (multi-dimensions or $\sigma > 1$), these are the only two and so we will focus on them.

Existence of the conserved energy H is of immense importance for understanding the behavior of solutions. Since it is the sum of two positive terms, neither of them can become larger than the initial value of H . This provides a bound on u and on its first derivative, which guarantees that the solution remains well-behaved for all time.

Focusing The *focusing* or *attractive* form of the equation is

$$i u_t = u_{xx} + |u|^2 u,$$

in which only the sign of the nonlinear term (relative to the derivative) has been changed. The particle number N is conserved as above. But now the conserved Hamiltonian is

$$H(t) = \int \left(|\nabla u|^2 - \frac{1}{2}|u|^4 \right) dx$$

with a sign change. Now the conserved energy is the *difference* of two positive terms. Nothing prevents both of them from becoming infinite together: the solution can and does develop very interesting singularities.

Splitting

A standard way to handle equations like NLS, in which u_t is the sum of two terms, is to look at the PDEs in which only one term appears at a time. Here the structure is particular simple, because each one is a type of rotation, one in physical and one in Fourier space.

Rotation in Fourier space First, let us consider the linear wave equation

$$i u_t = u_{xx}.$$

Since this is a linear equation with constant coefficients, its solutions are described by looking at individual Fourier modes. With $u(x, t) = U(t) \exp(i\xi x)$, we immediately see that $iU'(t) = -\xi^2 U(t)$, so we can write the solution as

$$u(x, t) = U_0 e^{i\xi^2 t} e^{i\xi x} = U_0 e^{i\xi(x+\xi t)}.$$

Of course, in a periodic box of length L , ξ is restricted to the discrete values $2\pi k/L$ for k an integer.

The phase speed $c = -\xi$ of the mode depends on ξ . Thus this wave equation is *dispersive*: different wavelengths propagate at different speeds. This is in contrast to hyperbolic systems like $u_{tt} = c^2 u_{xx}$, in which each different wavelength propagates at the *same* speed c (except for errors introduced by the discretization). Only for hyperbolic systems can an initial disturbance move while preserving shape. For a dispersive system any solution inevitably breaks up into a combination of many waves.

It is trivial to compute this solution in the Fourier representation. If

$$u(x, 0) = \sum_k \hat{u}_k(0) e^{2\pi i k x / L},$$

(whether the sum is finite or not), then the solution at later times is

$$u(x, t) = \sum_k \hat{u}_k(t) e^{2\pi i k x / L}, \quad \hat{u}_k(t) = e^{i(2\pi k / L)^2 t} \hat{u}_k(0).$$

Each mode just sits there and spins independently of all the others. Note that the rotation rate increases quadratically with k . On a discrete grid of size n , the highest mode is $n/2$, so the fastest rotation rate is $(\pi n / L)^2$. The rotation period of the k th mode is $T_k = 2\pi / (2\pi k / L)^2 = L^2 / 2\pi k^2$, so the shortest period is $T_{n/2} = 2L^2 / \pi n^2$.

If you wanted to do an explicit method, without using the special rotation structure, then you would need to take the time step τ smaller than this intrinsic time. It is just as for the diffusion equation, where we needed to resolve the decay time of the smallest modes. We could have avoided the problem there by doing a Fourier transform and evolving each mode separately, as we are doing here (but the FFTs would have been much slower than the finite differencing).

Rotation in physical space The other half of the problem is

$$i u_t = -|u|^2 u.$$

Since this equation has no space derivatives, it is just an ODE at each point, and the solution is immediately seen to be

$$u(x, t) = e^{i|u|^2 t} u(x, 0).$$

Again, this is just a rotation, that keeps constant the value of $|u(x, t)|^2$. Each separate spatial point just sits and spins independently.

Pseudo-spectral algorithm

Now we can put all this together into a numerical algorithm. The idea is that we do each effect in alternation. But since one is easiest to do in physical space, the other in Fourier, we have to keep transforming back and forth. That is why it is called “pseudo-spectral:” a fully spectral algorithm would do all operations in Fourier space, but that works for very few problems since almost everything has nonlinearity. Such algorithms only became possible with the invention of the fast Fourier transform.

Here is the outline. We start with a list $(u_0^m, \dots, u_{n-1}^m)$ representing the solution at time level m (keeping n for the number of space grid points). Here’s how we get to level $m + 1$ with time step τ .

1. *Forward transform.* Do an FFT on u^m to compute the amplitudes \hat{u}_k^m in the Fourier-space representation.
2. *Fourier rotation.* Spin each mode as follows:

$$\hat{u}_k^{m+1/2} = \exp(i\xi_k^2\tau) \hat{u}_k^m, \quad \xi_k = \frac{2\pi}{L} \min\{k, n - k\}$$

(it should be $k - n$ rather than $n - k$, but we square it anyway).

3. *Backward transform.* Do an inverse FFT on $\hat{u}^{m+1/2}$ to compute the elements $u_j^{m+1/2}$ in the physical-space representation.
4. *Physical rotation.* Rotate each grid point in place:

$$u_j^{m+1} = \exp\left(i\left|u_j^{m+1/2}\right|^2\tau\right) u_j^{m+1/2}$$

Let’s compute!

In the following, we will use Richardson Extrapolation. This is a method of accelerating a computation that’s done on an equal-spaced mesh. In its most general form, assume that $A(k)$ is the output that you get if you do the computation with time-step k and assume that $A(k/2)$ is the output that you get with time-step $k/2$. If the error is $\mathcal{O}(k^n)$ then we use n to create a new output:

$$\frac{2^n}{2^n - 1}A(k/2) - \frac{1}{2^n - 1}A(k)$$

This output will be more accurate than either of the parts that went into it.

We consider three time-stepping schemes:

1. The first scheme is as described above: given u^n , take one time step of size k to find u^* , the solution of $iu_t = u_{xx}$ with initial data u^n . Then take a time step of size k to find u^{n+1} , the solution of $iu_t = -|u|^n u$ with initial data u^* . This is the most basic splitting method.
2. The second scheme is a Strang splitting. Given u^n , take one time step of size $k/2$ to find u^* , the solution of $iu_t = u_{xx}$ with initial data u^n . Then take a time step of size k to find u^{**} , the solution of $iu_t = -|u|^n u$ with initial data u^* . Then take one time step of size $k/2$ to find u^{n+1} , the solution of $iu_t = u_{xx}$ with initial data u^{**} .
3. The third scheme is accelerated Strang splitting. Let u_c^{n+1} be what you get by applying the Strang splitting to u^n . Specifically, you took three time steps $k/2$, k , and $k/2$ to get from u^n to u_c^{n+1} . Now, let u_f^{n+1} be what you get by applying the Strang splitting twice. Specifically, you take steps $k/4$, $k/2$, and $k/4$ to get from u^n to $u^{n+1/2}$ and then take steps $k/4$, $k/2$, and $k/4$ to get from $u^{n+1/2}$ to u_f^{n+1} . We use Richardson extrapolation to blend the coarse and fine computations:

$$u^{n+1} = \frac{4}{3}u_f^{n+1} - \frac{1}{3}u_c^{n+1}$$

By construction, the first scheme will conserve the discrete L^2 norm of the mass

$$\sum_{j=0}^{N-1} |u_j^n|^2.$$

It will be first-order accurate in time. The second scheme will also conserve the mass and will be second-order accurate in time. The third scheme will be fourth-order accurate in time but it will not preserve the mass. Indeed, we know that

$$\sum_{j=0}^{N-1} |u_{f,j}^{n+1}|^2 = \sum_{j=0}^{N-1} |u_j^n|^2 \quad \text{and} \quad \sum_{j=0}^{N-1} |u_{c,j}^{n+1}|^2 = \sum_{j=0}^{N-1} |u_j^n|^2.$$

And so

$$\begin{aligned}
\sum_{j=0}^{N-1} \left| \frac{4}{3} u_{f,j}^{n+1} - \frac{1}{3} u_{c,j}^{n+1} \right|^2 &= \frac{16}{9} \sum_{j=0}^{N-1} |u_{f,j}^{n+1}|^2 + \frac{1}{9} \sum_{j=0}^{N-1} |u_{c,j}^{n+1}|^2 - \frac{8}{9} \sum_{j=0}^{N-1} \operatorname{Re} \left(u_{f,j}^{n+1} \overline{u_{c,j}^{n+1}} \right) \\
&= \frac{17}{9} \sum_{j=0}^{N-1} |u_j^n|^2 - \frac{8}{9} \sum_{j=0}^{N-1} \operatorname{Re} \left(u_{f,j}^{n+1} \overline{u_{c,j}^{n+1}} \right) \\
&= \sum_{j=0}^{N-1} |u_j^n|^2 + \frac{8}{9} \sum_{j=0}^{N-1} |u_j^n|^2 - \operatorname{Re} \left(u_{f,j}^{n+1} \overline{u_{c,j}^{n+1}} \right)
\end{aligned}$$

The second sum would need to equal zero for the mass to be preserved.

For all three schemes we also compute the mass $M(t_n)$ and the Hamiltonian $H(t_n)$ at each time step. Here

$$M(t_n) \sim \int_0^{2\pi} |u(x, t_n)|^2 dx, \quad H(t_n) \sim \int_0^{2\pi} |u_x(x, t_n)|^2 - \frac{1}{2}|u|^4 dx,$$

and the integrals are approximated via the trapezoidal rule. None of the three schemes conserve the Hamiltonian and so we will be able to use the Hamiltonian as another measure of accuracy. For this reason, whatever quadrature rule we use to approximate the mass and the Hamiltonian should be very accurate — we want to be testing the accuracy of the scheme, not of the quadrature rule. In general, the trapezoidal rule is a second-order accurate quadrature rule, however it's spectrally accurate for integrals of *periodic* functions.

We want to compare the schemes against exact solutions. The defocussing equation $iu_t = u_{xx} - |u|^2 u$ has the plane wave solution

$$Ae^{i(Bx+t(A^2+B^2)+C)} \quad (5)$$

and the focussing equation $iu_t = u_{xx} + |u|^2 u$ has the plane wave solution:

$$Ae^{i(Bx+t(B^2-A^2)+C)}, \quad (6)$$

where A , B , C , and D are any real number.

In addition, the focussing equation has soliton solutions

$$u(x, t) = \sqrt{2\alpha} \frac{\exp \left(i \left(\frac{U}{2} x - (\alpha - U^2/4) t + \phi_0 \right) \right)}{\cosh(\sqrt{\alpha}(x + Ut - x_0))} \quad (7)$$

The parameters $\alpha > 0$, U , ϕ_0 , and x_0 are arbitrary. These solutions are localised in space — the plot of $|u(x, t)|$ looks somewhat Gaussian. The maximum of $|u|$ is initially at $x = x_0$ and moves with speed U .

We start by testing the scheme for the defocussing equation with the plane wave solution:

$$u(x, t) = 4e^{i(3x-7t)} \quad \text{on } [0, 2\pi] \quad (8)$$

That is, we're taking the parameters in the exact solution (6) to be $A = 4$, $B = 3$, $C = 0$. We take $h = 2\pi/32$ and compute with the initial time step $k = 1/2$. We divide the time steps by 2 at each point. We monitor the L^∞ norm of the error at time $t_f = 5$ as well as how well the mass $|M(5) - M(0)|$ and Hamiltonian $|H(5) - H(0)|$ are preserved. We find:

k	$\ err\ _{L^\infty}$	$ M(5) - M(0) $	$ H(5) - H(0) $
1/2	4.4274e-03	4.2633e-14	6.4328e-04
1/4	1.1285e+01	1.9895e-13	8.5858e+03
1/8	1.0087e+01	4.4054e-13	9.0883e+03
1/16	9.9500e+00	8.5265e-14	7.4889e+03
1/32	8.9609e+00	1.9185e-12	8.6124e+03
1/64	2.9204e-12	2.1032e-12	5.2751e-11
1/128	6.1997e-12	3.9506e-12	9.8908e-11

We see that the mass is well-preserved for all choices of time step. This is unsurprising — mass conservation was built in to the scheme. However, the L^∞ error of the solution at time $t = 5$ is very poor for time steps $1/2 \dots 1/32$ and is excellent for time steps $1/64$ and $1/128$. We observe a similar thing for the Hamiltonian. This sharp transition from “lousy” to “fantastic” leads us to suspect there is an instability, which we observe in Figure 1. Here, we see that for time step $k = 1/32$ there is a numerical instability. The exact solution would have a single peak located at wave number -3 . This instability is gone for time-step $k = 1/64$. Because this stability constraint on the time-step is in the first time-stepping scheme (#1 on the above list) it is also in the other two — they are based on the first scheme.

We now test the scheme for the focussing equation with the plane wave solution:

$$u(x, t) = 4e^{i(3x+25t)} \quad \text{on } [0, 2\pi] \quad (9)$$

That is, we're taking the parameters in the exact solution (5) to be $A = 4$, $B = 3$, $C = 0$. We take the exact same parameters as for Figure 1. Here we observe something quite different when looking at the same time-steps

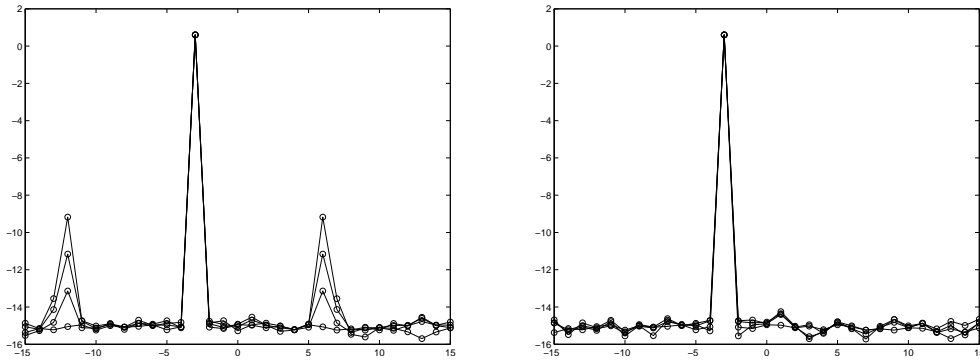


Figure 1: The spectrum is shown for the numerical approximation of the solution (8) at times $t = 0$, $t = 10/32$, $t = 20/32$, and $t = 30/32$. Left plot: The time step is $k = 1/32$. Right plot: The time step is $k = 1/64$.

as in Figure 1. In the left plot of Figure 2 we see the main peak in the spectrum centered at $\ell = -3$. We also see some of the wave numbers near $\ell = -3$ growing up from the level of round-off. And we see the wave numbers $\ell = -14, 8$ are also growing. In the right plot of Figure 2, we see that the growth at wave numbers $\ell = -14, 8$ has been suppressed but the growth in the wave numbers flanking the plane wave frequency $\ell = -3$ are still growing. In Figure 3, we plot the spectrum at a fixed time $t = 480/512$ for four computations done with smaller and smaller time steps. We see that the spectra essentially overlap. This suggests that the behaviour near $\ell = -3$ reflects a real instability rather than a numerical instability. That is; if one were to linearize the focussing Schrödinger equation about the plane wave solution (9) then one will find unstable directions and eigenvalues with positive real parts. Indeed, one can do this and find there is an instability. This particular instability was discovered by Benjamin and Feir and is called the Benjamin-Feir instability or the modulation instability.

Now we would like to test the code against the soliton solutions (7). Here, we will have some difficulties. First of all, the soliton solution is a “one-hump” solutions on \mathbb{R} . It is not periodic in space — if you plot $|u|$ you do not see an infinite array of bumps. In this case, it’s natural to try and take the domain much larger than the size of the bump — because $|u|$ decays quickly to zero, if the domain is large enough then $|u|$ will be nearly zero at both ends of the interval (which would then make u look periodic). And while the bump is moving with finite speed U , it would be a while

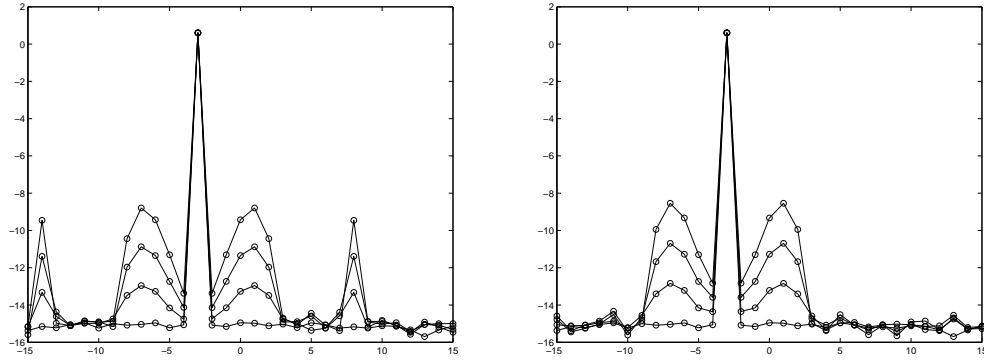


Figure 2: The spectrum is shown for the numerical approximation of the solution (9) at times $t = 0$, $t = 10/32$, $t = 20/32$, and $t = 30/32$. (Left plot: The time step is $k = 1/32$. Right plot: The time step is $k = 1/64$.)

before this is felt at the endpoints of the interval — for some period of time the solution would continue to “look periodic”. Let’s see how this works in practice.

Consider the solution

$$u_{sol}(x, t) = \sqrt{400} \frac{\exp\left(i\left(\frac{1}{2}x + (40000 - \frac{1}{4})t\right)\right)}{\cosh(\sqrt{200}(x - t - \pi))} \quad (10)$$

This is the exact solution (7) with parameters $\alpha = 200$, $U = 1$, $x_0 = \pi$, and $\phi_0 = 0$. We consider this solution for times between $t = 0$ and $t = 1$. At time $t = 0$, we find at the endpoints $|u_{sol}(0, 0)| = 2.0e - 18$ and $|u_{sol}(2\pi, 0)| = 2.0e - 18$. At time $t = 1$, we find $|u_{sol}(0, 1)| = 2.8e - 12$ and $|u_{sol}(2\pi, 1)| = 1.5e - 24$. Note that at $x = 2\pi$ the exact solution is already a little above the level of round-off. This shows up in the spectrum, see the left plot of Figure 4, where we see that the spectrum for the exact solution at time $t = 1$ flattens out at a slightly higher value. In the right plot of Figure 4, we present the spectra of the solutions as computed using the first scheme. The time step is $k = 1/2000$ and the solutions were computed up to time $t = 1$. We see that the spectrum of the approximate solution agrees fairly well with the exact solution for wave numbers $\ell = -60 \dots 60$ but then deviates markedly. In Figure 5 we present the analogous plots as in Figure 4 except we look at the plot of $|u|$. The left plot shows the magnitude of the exact solution (10) at time $t = 0$ and the solid line shows this magnitude at time $t = 1$. The right plot shows the result of the computation using the first scheme and $N = 2048$ points and time steps $k = 1/2000$. The

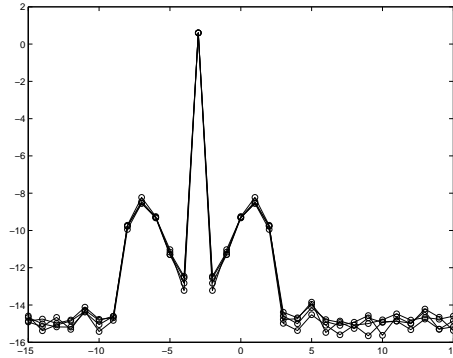


Figure 3: The spectrum is shown for the numerical approximation of the solution (9) at time $t = 480/512$. The spectra are shown from four runs with time steps $k = 1/64$, $k = 1/128$, $k = 1/256$, and $k = 1/512$. The graphs overlap sufficiently that we don't try to distinguish them.

magnitude of the initial data is plotted with the dashed line and the magnitude of the approximate solution at time $t = 1$ is plotted with a solid line. Note that there are oscillatory pulses in the regions where the exact solution is very close to zero. These pulses do get smaller as the time step is refined however it becomes very CPU-intensive. I'm not an expert in computing solutions, but the above numerical experiments make me suspect that naive spectral methods are not the way to go. I would consult with an expert before proceeding any further.

None of the above was testing the three schemes head-to-head. We now do this with initial data for which we don't have an exact solution. We consider the focussing case of the equation and take the initial data

$$u_0(x) = \exp(e^{ix}).$$

We take $h = 2\pi/128$ and compute up to time $t_f = 1/10$. The coarsest time-step is $k = 1/100$ and we compute seven approximate solutions, dividing k by two with each computation. Call these solutions u_1, u_2, \dots, u_7 . As before, we monitor the quantities:

$$\|u_j(\cdot, 1/10) - u_{j+1}(\cdot, 1/10)\|_{L^\infty}, \quad |M(1/10) - M(0)|, \quad |H(1/10) - H(0)|$$

For the first scheme we find:

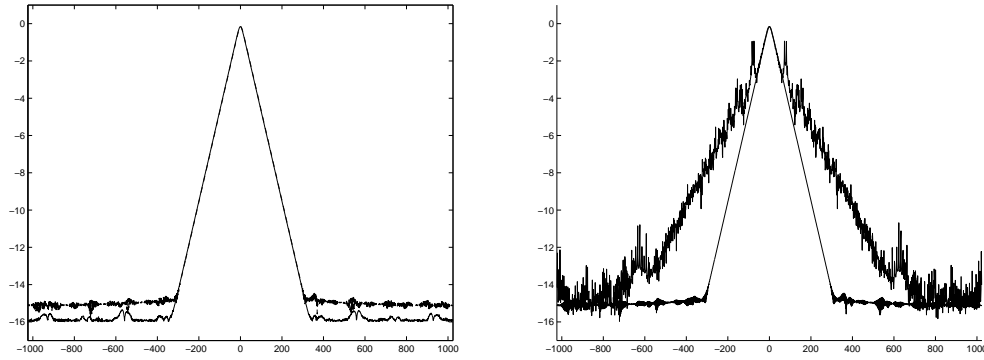


Figure 4: Left plot: the spectrum for the exact solution (10) shown at times $t = 0$ and $t = 1$. The solution has been sampled at 2048 points. The spectra overlap for amplitudes greater than $10e - 12$. The solution at time $t = 0$ has a spectrum that “flattens out” at $10e - 16$, the solution at time $t = 1$ has a spectrum that “flattens out” at $10e - 15$. Right plot: The three schemes are used to compute approximate solutions up to time $t = 1$. The time step is $k = 1/2000$. The spectrum of the exact solution at time $t = 1$ is shown as a guide line.

k	$\ u_j - u_{j+1}\ _{L^\infty}$	ratios	$ M(5) - M(0) $	$ H(5) - H(0) $	ratios
1/100	6.3721e-03	2.0015	-8.8818e-15	2.1151e-01	2.0218
1/200	3.1837e-03	2.0003	2.4869e-14	1.0461e-01	2.0107
1/400	1.5916e-03	2.0000	1.4211e-14	5.2026e-02	2.0053
1/800	7.9581e-04	2.0000	7.1054e-15	2.5944e-02	2.0027
1/1600	3.9791e-04	2.0000	1.7586e-13	1.2955e-02	2.0013
1/3200	1.9896e-04		8.8818e-15	6.4731e-03	2.0007
1/6400			1.5632e-13	3.2355e-03	

We see that the ratios of $\|u_j - u_{j+1}\|$ are going to 2 as expected. The mass is conserved, as expected. The Hamiltonian is not conserved but as the time step gets smaller the Hamiltonian is more approximately conserved. Note that the Hamiltonian can also be used to test the convergence of the scheme; the ratios of the amount of drift are going to 2.

We now test the second scheme, which is simply the splitting scheme made more accurate via Strang splitting:

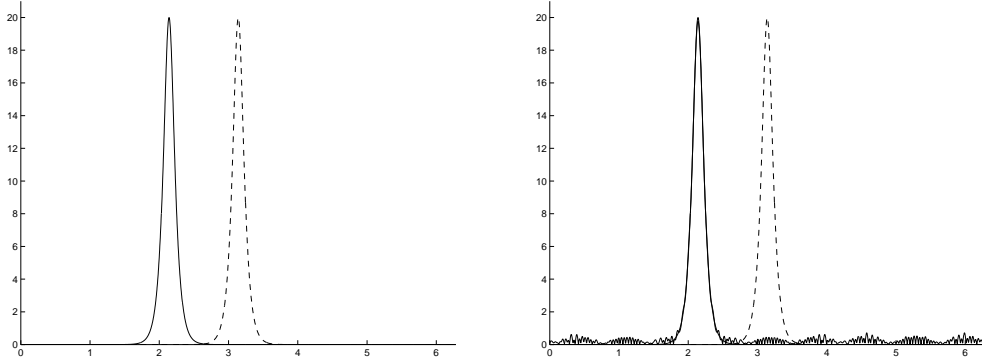


Figure 5: Left plot: The dashed line is the magnitude of the exact solution $|u_{sol}|$ plotted at time $t = 0$. The solid line is the magnitude at time $t = 1$. Right plot: We use the first scheme to compute up to time $t = 1$ using $N = 2048$ points and time steps of size $k = 1/2000$. The dashed line is the magnitude of the initial data ($|u_{sol}|$) at time $t = 0$. The solid line is the magnitude of the approximate solution at time $t = 1$.

k	$\ u_j - u_{j+1}\ _{L^\infty}$	ratios	$ M(5) - M(0) $	$ H(5) - H(0) $	ratios
1/100	4.4332e-04	4.0047	0	2.5412e-03	4.0088
1/200	1.1070e-04	4.0012	2.6645e-14	6.3391e-04	4.0022
1/400	2.7667e-05	4.0003	2.1316e-14	1.5839e-04	4.0005
1/800	6.9163e-06	4.0001	1.2790e-13	3.9592e-05	4.0001
1/1600	1.7290e-06	4.0000	1.7764e-14	9.8977e-06	4.0000
1/3200	4.3226e-07		1.8296e-13	2.4744e-06	4.0000
1/6400			1.3323e-13	6.1860e-07	

We see that the ratios of $\|u_j - u_{j+1}\|$ are going to 4 as expected. The mass is conserved, as expected. Again, the Hamiltonian is only approximately conserved. As before, the Hamiltonian can also be used to test the convergence of the scheme; the ratios of the amount of drift are going to 4.

Finally, we test the third scheme which is Strang splitting accelerated with Richardson Extrapolation:

k	$\ u_j - u_{j+1}\ _{L^\infty}$	ratios	$ M(5) - M(0) $	ratios	$ H(5) - H(0) $	ratios
1/100	2.8869e-07	16.432	1.2370e-08	32.236	1.3443e-06	15.977
1/200	1.7568e-08	16.197	3.8372e-10	31.609	8.4141e-08	15.988
1/400	1.0846e-09	16.083	1.2140e-11	56.479	5.2628e-09	15.979
1/800	6.7440e-11	16.172	2.1494e-13	5.5000	3.2936e-10	18.646
1/1600	4.1701e-12	11.489	3.9080e-14	.21569	1.7664e-11	56.500
1/3200	3.6295e-13		1.8119e-13	1.4783	3.1264e-13	.061111
1/6400			1.2257e-13		5.1159e-12	

We see that the ratios of $\|u_j - u_{j+1}\|$ are going to 16. These ratios “get bad” once the difference is near round-off error. The mass is not conserved but as the time step gets smaller the mass is more approximately conserved. When we use the mass to test the convergence we see ratios that are close to 32. These ratios “get bad” once the drift is near round-off error. Again, the Hamiltonian is only approximately conserved. As before, the Hamiltonian can also be used to test the convergence of the scheme; the ratios of the amount of drift are going to 16.